IEEE/RSJ International Workshop on Intelligent
Robots and Systems IROS '91, Nov. 3-5, 1991,
Osaka, Japan. IEEE Cat. No. 91TH0375-6

Hyper-Redundant Robot Mechanisms and Their Applications

Gregory S. Chirikjian Joel W. Burdick

School of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125, USA

Abstract:

Hyper-redundant robots have a large or infinite number of degrees of freedom. Such robots are analogous to snakes or tentacles and are useful for operation in highly constrained environments and novel forms of locomotion. This paper reviews newly developed methods for the kinematic analysis of hyper-redundant manipulators. These methods can be applied to a wide variety of hyper-redundant morphologies and lead to very efficient inverse kinematic, path planning, obstacle avoidance, locomotion, and grasping schemes. This paper also reviews the design and implementation of a planar 30 degree of freedom variable geometry truss hyper-redundant robot.

1. Introduction

Hyper-redundant manipulators have a relative degree of kinematic redundancy which is large or infinite. In operation, these manipulators approximate the shape of snakes, elephants' trunks, or tentacles. These robots have been investigated in varying forms for more than a quarter of a century. Hyper-redundant manipulators have previously been called 'swan's neck' [8], 'highly articulated' [13], and 'tentacle' [14] among a variety of other names. The term 'hyper-redundant,' coined by the authors in [3], is used to describe this broad class of robotic devices, such as those described in [1,2-7,10,11,17]. Because of their highly articulated nature, such robots may be used for inspection and operation in highly constrained environments [2,3]. Further, in analogy with naturally occurring animals, hyper-redundant robots can also be used to implement novel means of locomotion and grasping [5].

The mechanical implementation of a hyper-redundant manipulator may take many forms, depending upon the requirements of a particular application. Hyper-redundant robots may consist of a large number of rigid links (Figure 1(a)). Alternatively, these robots may consist of a truly flexible structure, such as pneumatically driven arms [16,17], as depicted in Figure 1(b). Variable geometry truss manipulators (or VGTMs) [11,13,15] are another possible morphology (Figure (1c)). Several other design concepts for hyper-redundant manipulators can be found in the literature [1,9,10].

Algorithms for redundant manipulator inverse kinematics typically require the computation of a Jacobian pseudo-inverse [12]. This computation becomes prohibitive as the number of manipulator degrees of freedom increases, and can be impractical for hyper-redundant robots. Until recently, the lack of an efficient and unified framework for kinematic analysis and motion planning of these robots has been one significant stumbling block to their widespread practical implementation.

In [2-7, and references therein], the authors have introduced novel methods for kinematic analysis which does not rely upon the manipulator Jacobian matrix. These methods can be applied to a wide variety of hyper-redundant mechanical morphologies and form the basis for efficient manipulation, obstacle avoidance, locomotion, and grasping schemes. Our analysis of the direct and inverse kinematics of both *extensible* (variable length) and *nonextensible* (fixed-length) hyper-redundant manipulators is based on an intrinsic differential geometric modeling of a 'backbone curve,' which captures the macroscopic geometric features of the robot. These techniques were used in [3] to implement an efficient obstacle avoidance scheme for hyper-redundant robots. Other authors have developed non-intrinsic methods for resolving the hyper-redundancy of variable geometry truss manipulators (VGTMs) [13,15], but it is not clear if these methods can be applied to other hyper-redundant manipulator morphologies.

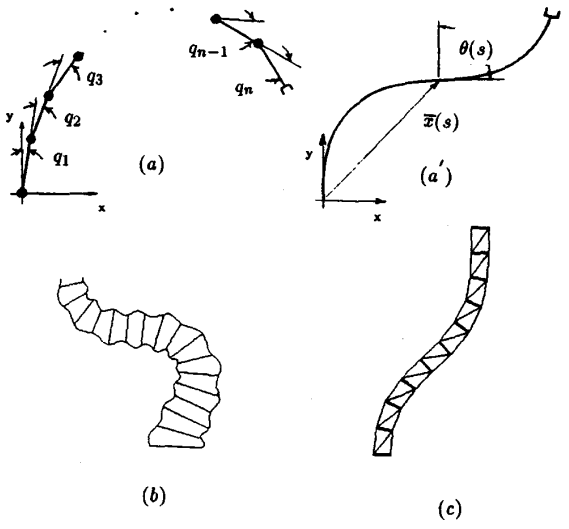

Figure 1: Some Hyper-Redundant Robot Morphologies: Discrete Link, Pneumatic, and VGT

Because of length restrictions, this paper focuses only on planar hyper-redundant systems and one particular implementation morphology, the planar VGT. Section 2 reviews the kinematic analysis of hyper-redundant manipulators presented in [3-5], and extends the analysis to nondifferentiable curves. Section 3 reviews the modal approach to hyper-redundancy resolution for nonextensible manipulators in unconstrained environments, and presents new closed form inverse kinematic solutions. Section 4 illustrates hyper-redundant manipulator obstacle avoidance in

the case of a moving obstacle field. Section 5 illustrates concepts for hyper-redundant robot grasping and manipulation of objects formulated as single and dual arm tasks. The examples elaborate on the concept of the 'grasping wave' for the fine manipulation of objects, as in [5], and extend the formulation to include dual arm tasks. Finally, Section 6 describes an actual 30 degree-of-freedom planar hyper-redundant VGTM robot under development by the authors.

2. Intrinsic Formulation of Planar Hyper-Redundant Manipulator Kinematics

We motivate the kinematic modeling approach as follows. Consider the planar n-link manipulator in Figure 1(a), whose k^{th} link has length l_k. The position of the $k+1^{st}$ joint of the manipulator, as measured in a base frame, is denoted by $\overline{x}_d^k = [x_d^k, y_d^k]^T$. The orientation of the k^{th} link measured *clockwise* from the base frame y-axis is denoted by Q_d^k. It follows that:

$$x_d^k = \sum_{i=1}^{k} l_i \sin Q_d^i; \qquad y_d^k = \sum_{i=1}^{k} l_i \cos Q_d^i; \tag{1}$$

where $Q_d^k = \sum_{j=1}^{k} q_j$, and $\{q_i\}$ are clockwise measured joint angles. Note that $\{l_i\}$ and $\{q_i\}$ may vary with time depending on whether prismatic or revolute actuators (or both) are used at each link. As n approaches infinity, the manipulator shape approximates a continuous curve, as represented in Figure 1(a').

While a curve is an exact representation of a manipulator with an infinite number of links, the relevant features of manipulators with a finite but large number of degrees of freedom can also be specified by a *backbone curve* which lies along the robot. Kinematic analysis and task programming is thus reduced to the study of the spatial and temporal behavior of the backbone curve.

Kinematic solutions based on the backbone curve can be used directly to control the geometry of continuous hyper-redundant manipulators, while an additional step, called the 'fitting process' [3,6], is required for the continuous solution to be used to determine the actuator displacements for discrete geometry systems.

Let $\overline{x}_c(s,t) = [x_c(s,t), y_c(s,t)]^T$ denote the position vector to any point on a backbone curve parameterized by $s \in [0,1]$, at time t. $x_c(s,t)$ and $y_c(s,t)$ are given by:

$$x_c(s,t) = \int_0^s l_c(\sigma,t)\sin\theta_c(\sigma,t)d\sigma \tag{2a}$$

$$y_c(s,t) = \int_0^s l_c(\sigma,t)\cos\theta_c(\sigma,t)d\sigma \tag{2b}$$

where $l_c(s,t)$ is a function which defines the length of the backbone curve tangent at point s. $l_c(s,t)$ is the rate of change of backbone curve arc length with respect to parameter s, i.e., arclength of the backbone curve is:

$$L(s,t) = \int_0^s l_c(\sigma,t)d\sigma. \tag{3}$$

$\theta_c(s,t)$ is the clockwise measured angle which the tangent to the curve at the point s makes with the y-axis:

$$\theta_c(s,t) = \int_0^s \kappa(\sigma,t) l_c(\sigma,t)d\sigma \tag{4}$$

The *curvature function*, $\kappa(s,t)$, specifies how the curve bends at each point, s, and is defined as the magnitude of the second derivative of the position vector with respect to arc length. Thus,

$$\frac{\partial \overline{x}_c}{\partial L} = \frac{\partial \overline{x}_c}{\partial s}\frac{ds}{dL} = \frac{\partial \overline{x}}{\partial s}\frac{1}{l_c} = (\sin\theta_c, \cos\theta_c)^T,$$

and so

$$\kappa^2(s,t) = \frac{\partial^2 \overline{x}_c}{\partial L^2}\cdot\frac{\partial^2 \overline{x}_c}{\partial L^2} = \frac{1}{l_c^2}\left(\frac{\partial\theta_c}{\partial s}\right)^2$$

is defined in the same way as the classical definition, taking into account extensibility. Hereafter we denote the end-effector position by $\overline{x}_{ee} = \overline{x}_c(1,t)$, or $x_{ee}(t) = x_c(1,t)$, $y_{ee}(t) = y_c(1,t)$. Similarly, the end-effector orientation is denoted $\theta_{ee}(t) = \theta_c(1,t)$. The functional form of $\overline{x}_c(s,t)$ can be used to directly control the geometry of a flexible morphology manipulator. For discretely segmented structures, the backbone curve is used with a fitting process to determine the discrete actuator displacements such that the actual manipulator conforms to the backbone curve geometry [3,6].

We extend the analysis methods in [3-6] to include nondifferentiable curves by allowing the curvature function to be piecewise continuous with a finite number of jumps and integrable singularities. Curvature functions of this class can be represented by:

$$\kappa(s,t) = \sum_{i=1}^{m} \kappa_i(s,t) W(s, s_{i-1}(t), s_i(t)) + \sum_{j=1}^{m_d} \theta_j \delta(s-\sigma_j(t)) \tag{5}$$

where $\{s_i(t), \sigma_i(t)\} \in [0,1]\times[0,1]$. $\kappa_i(s,t)$ is a continuous function on the interval $(s,t) \in [s_{i-1}, s_i]\times[t_0,t_1]$ and

$$W(s, s_{i-1}(t), s_i(t)) = H(s - s_{i-1}(t)) - H(s - s_i(t))$$

is a 'window function' which is the difference between two Heaviside unit step function, and m is the number of jump discontinuities. θ_j and σ_j are respectively the strength and location of the j^{th} impulsive singularity, $\delta(s)$ is the Dirac delta function, and m_d is the number of such singularities. It is interesting to note that a curvature function and rate of arc length function of the form

$$\begin{aligned} \kappa(s,t) &= \frac{1}{l_c(s,t)}\sum_{i=1}^{n}\theta_d^i(t)\delta\left(s - \frac{i-1}{n}\right) \\ l_c(s,t) &= n\sum_{i=1}^{n} l_i(t) W\left(s, \frac{i-1}{n}, \frac{i}{n}\right) \end{aligned} \tag{6}$$

will cause the continuous curve in figure 1(a') to become the discrete manipulator shown in Figure 1(a). Effectively, a delta-function in curvature with variable strength θ corresponds to a revolute joint with joint angle θ.

A curvature function with delta function singularities will yield manipulator backbone curves which are continuous but not differentiable at the points where the delta function singularities are defined. This can cause problems in the application of the fitting method in [6]. For example, if the manipulator physical structure 'slides' along a backbone curve, as may be the case in an obstacle avoidance situation [3], an instantaneous change in orientation is specified when the manipulator reaches a delta function in curvature. To overcome this difficulty, we introduce a *smoothed orientation function*, $\theta_c^*(s,t)$. This is demonstrated by example in the next section.

3. A Modal Approach to Hyper-Redundancy Resolution and Inverse Kinematics

The forward kinematics of a planar backbone curve can be computed by exact or numerical integration of (2,4). The inverse kinematic solution in this case can be interpreted as the choice of a backbone curve which satisfies end-effector position constraints, and possibly other task constraints. For a given problem, there are an infinite number of possible backbone shapes, and an optimality criterion can be used to select an optimal shape

[7]. In this section we propose a restricted class of solutions which have a useful engineering property: they are computationally efficient. In this 'modal approach', θ_c and l_c are restricted to have a modal form:

$$\theta_c(s,t) = \sum_{i=1}^{N_\theta} a_i(t)\Phi_i(s), \tag{7a}$$

$$l_c(s,t) = \sum_{i=1}^{N_l} a_{\{i+N_\theta\}}(t)\Phi_{\{i+N_\theta\}}(s) \tag{7b}$$

where Φ_i is a *mode function*, and a_i is a *modal participation factor* [2-6]. N_θ is the number of bending modes. N_l is the number of extension modes, or longitudinal degrees of freedom along the backbone curve. A hyper-redundant mechanism constrained to these modes effectively has $N = N_\theta + N_l$ degrees of freedom. N is chosen large enough to satisfy the number of end-effector or other task constraints. With this restriction, the end-effector coordinates and mechanism shape become a function of the $\{a_i\}$. The $\{\Phi_i\}$ are user-specified functions, which can vary with task requirements, or even within a task. The restriction to modal form reduces the inverse kinematic problem to the search for the $\{a_i\}$ which satisfy task constraints. When N equals the number of constraints, inverse kinematic solutions based on (7) serve as a means of 'hyper-redundancy resolution'. Analogous methods hold for spatial backbone curves [4,6]. Note that limitations on a mechanism's ability to bend and/or extend and contract impose bounds on $\{a_1, a_2, ..., a_N\}$ which are analogous to joint limits in traditional manipulators.

For some choices of modes, exact closed form inverse kinematic solutions can be found. One choice of modes which was investigated in [3] for the case $N_\theta = 2$, $N_l = 0$ is:

$$\Phi_1(s) = \sin 2\pi s; \qquad \Phi_2(s) = 1 - \cos 2\pi s \tag{8}$$

with $l_c(s,t) = 1$. Substituting (8) into (7) and then (2), and evaluating at $s = 1$, it can be shown that the forward kinematics equations reduce to

$$x_{ee} = \sin(a_2) J_0\left[(a_1^2 + a_2^2)^{\frac{1}{2}}\right] \qquad y_{ee} = \cos(a_2) J_0\left[(a_1^2 + a_2^2)^{\frac{1}{2}}\right] \tag{9}$$

where J_0 is the zeroth order Bessel function. The 'inverse kinematics' (evaluation of modal participation factors) in this case can be computed as:

$$a_1 = \hat{a}_1^{\pm}(\overline{x}_{ee}) = \pm\left(\left[J_0^{-1}\left[(x_{ee}^2 + y_{ee}^2)^{\frac{1}{2}}\right]\right]^2 - [\mathrm{Atan2}(x_{ee}, y_{ee})]^2\right)^{\frac{1}{2}} \tag{10a}$$

$$a_2 = \hat{a}_2(\overline{x}_{ee}) = \mathrm{Atan2}(x_{ee}, y_{ee}). \tag{10b}$$

J_0^{-1} is the 'restricted inverse Bessel function of zero order', and is defined as the inverse of $J_0(x)$ for $0 < x < \mu$ where $\mu \approx 3.832$ is the first local minimum of J_0. See [2,4] for details and additional closed form solutions. The plus and minus sign distinguishes two possible poses. Other poses would exist if other intervals of the argument were permitted in computing the inverse of $J_0(x)$.

A nonsmooth example is the choice of modes for $N = N_\theta = 3$, $N_l = 0$:

$$\Phi_1(s) = H(s - L_1); \Phi_2(s) = H(s - L_2); \Phi_3(s) = H(s - L_3) \tag{11}$$

with $l_c(s,t) = 1$, where $0 < L_1 < L_2 < L_3 < 1$ are specified numbers. Substituting these modes into (7) and (2) and evaluating at $s = 1$, the forward kinematics equations reduce to

$$x_{ee} = \mathcal{L}_1 \sin a_1 + \mathcal{L}_2 \sin(a_1 + a_2) + \mathcal{L}_3 \sin(a_1 + a_2 + a_3) \tag{12a}$$

$$y_{ee} = L_1 + \mathcal{L}_1 \cos a_1 + \mathcal{L}_2 \cos(a_1 + a_2) + \mathcal{L}_3 \cos(a_1 + a_2 + a_3) \tag{12b}$$

$$\theta_{ee} = a_1 + a_2 + a_3 \tag{12c}$$

where $\mathcal{L}_1 = L_2 - L_1, \mathcal{L}_2 = L_3 - L_2$, and $\mathcal{L}_3 = 1 - L_3$. This is nothing more than the forward kinematics of a three-link planar revolute manipulator which is translated along the y-axis by L_1, and with joint angles $\{a_1, a_2, a_3\}$ and link lengths $\{L_2 - L_1, L_3 - L_2, 1 - L_3\}$. The 'inverse kinematics' (evaluation of modal participation factors) in this particular case can then be computed in the usual way.

In Figure 2 (and all subsequent figures) a variable geometry truss is used to illustrate these concepts, though these methods are applicable to other implementation morphologies (with the application of appropriate fitting algorithms). Figure 2 shows three different sets of modes of the form given by (11) (corresponding to different "link lengths") for the same end-effector location. The lengths in the figure are $\{L_1, L_2, L_3\} = \{0.1, 0.3, 0.9\}, \{0.1, 0.5, 0.8\}$, and $\{0.2, 0.5, 0.7\}$ and the end-effector position is $(x_{ee}, y_{ee}) = (0.3, 0.9)$.

In the usual fitting procedure for this mechanism morphology [6], which assumes a continuously differentiable backbone curve, the fixed truss elements are chosen to be normal to the backbone curve. However, when the backbone curve is not continuously differentiable, the normal is not well defined, and smoothing is used to define alternate orientations for the transverse truss elements. The function $\theta_c(s,t) = \sum_{i=1}^{3} a_i(t) H(s - L_i(t))$ in the above example was smoothed to form $\theta_c^*(s,t)$ by using a cubic polynomial to replace the value of the function $\theta_c(s,t)$ on the intervals $L_i - \epsilon < s < L_i + \epsilon$ for $i = 1,2,3$. In this way, a smooth transition results across the singularities while matching the original function outside an ϵ neighborhood of the singularities. ϵ is small compared to the smallest link length. While θ_c is used in all backbone curve computations, such as (2), θ_c^* is used for the fitting procedure.

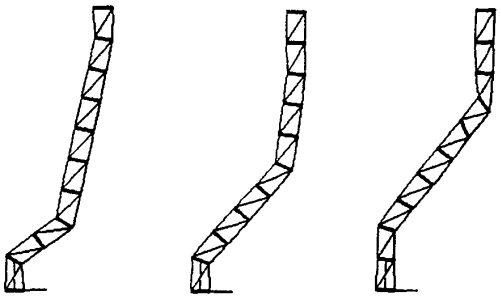

Figure 2: Nondifferentiable Curvature Modes

It is not necessary to choose modes which result in closed form inverse kinematic solutions as in (8) or (11) to use this method. Look-up tables or neural networks can be trained to store the mapping between participation factors and end-effector coordinates for any given set of manipulator modes. Thus the speed associated with closed form inverse kinematic solutions can be attained for modes which do not have closed form inverse kinematic solutions.

4. Hyper-Redundant Manipulator Obstacle Avoidance

Hyper-redundant manipulators are ideal for obstacle avoidance because of their ability to assume diverse and complicated geometries.

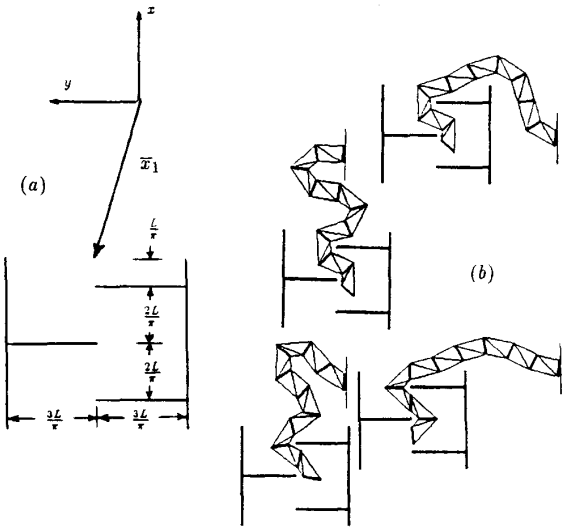

Figure 3: Obstacle Avoidance

Figure (3b) shows how a hyper-redundant manipulator can maneuver through the obstacle defined in Figure (3a). Obstacle avoidance can be effected by defining 'tunnels' through the highly constrained environment. The manipulator must 'slither' through these tunnels to reach its goal. For manipulator sections outside of the obstacle field, methods such as the modal approach presented in Section 3 can be used to specify the position of a point on the manipulator so that it is located at the entrance to the obstacle field. Inside the obstacle field, segments of the manipulator backbone curve assume values of $\theta_c(s,t)$ and $l(s,t)$ which are consistent with the local tunnel geometry. In the particular example shown in Figure 3, the manipulator is nonextensible: $l_c(s,t) = 1$. The associated backbone curvature function ($\kappa(s,t) = \partial\theta_c/\partial s$) is of the form:

$$\kappa(s,t) = \kappa_1(s,t)W(s,0,s_1(t)) + \kappa_2(s,t)W(s,s_1(t),1) \quad (13a)$$

where

$$\kappa_1(s,t) = \frac{2\pi}{s_1(t)}\hat{a}_1^+(\overline{x}_1/s_1)\cos\frac{2\pi s}{s_1(t)} + \frac{2\pi}{s_1(t)}\hat{a}_2(\overline{x}_1/s_1)\sin\frac{2\pi s}{s_1(t)} \quad (13b)$$

$$\kappa_2(s,t) = -\frac{\pi}{L}[W(s-s_1(t),0,L) - W(s-s_1(t),L,2L) + W(s-s_1(t),2L,3L)]. \quad (13c)$$

κ_1 assures that a point moving along the manipulator backbone at $s = s_1(t)$ has fixed position with respect to the obstacle field entrance provided the entrance is within reach. That is, κ_1 assures that $\overline{x}_c(s_1(t)) = \overline{x}_1(t)$, which is the position vector of the entrance to the obstacle field in the base frame. The functions $\hat{a}_i$ are defined in (10). κ_2 has the form of a traveling wave (composed of three semicircular arcs) for the section of the manipulator which is inside the tunnel. For details of the formulation which resulted in the above curvature functions, see [2,3].

5. Hyper-Redundant Grasping

This section investigates the kinematics of hyper-redundant grasping and introduces a grasping scheme based on a 'grasping wave.' This notion is based on a previous analysis of hyper-redundant locomotion [5]. Here we consider only *pure bending* waves, i.e., the manipulator is nonextensible. A more general formulation for extensible systems can be found in [2].

Object reorientation can be effected by the following sequence of operations:

- **Shape Initialization:** The hyper-redundant mechanism wraps around the object. The section of the manipulator in contact with the object is termed the *grasp contact segment*, and is specified by the backbone curve segment $s \in [s_g, 1]$. The remainder is termed the *noncontact segment*, and is specified by $s \in [0, s_g)$.
- **First Phase:** A section of the mechanism, $s \in [s_{w_0}, s_{w_1}]$, in the noncontact segment distorts to a wave form. As a result, the object will be displaced by a small amount. This phase is shown in Figure 4(a).
- **Second Phase:** The wave generated in the first phase travels along the mechanism toward the distal end without changing the position or orientation of the object over which it passes. This phase is shown in Figure 4(b). When the wave has traveled to the distal end of the manipulator, the grasp contact segment will be longer by an amount s_L, referred to as 'slack'.
- **Third Phase:** The manipulator 'unwraps' part of the grasp contact segment from the object by an amount s_L by straightening the mechanism in the section of the grasp contact segment which is closest to the noncontact segment: $s \in [s_g, s_g + s_L]$. This results in a rotation and displacement of the object.

When the third phase is complete, the cycle repeats starting with the first phase. This repetition results in repeated object rotations, the magnitude of which depend on the value of $L = s_{w_1} - s_{w_0}$, and the slack, s_L. When the manipulated object is a disk, the displacements resulting from the first and third phases cancel, leaving only a net rotation. For arbitrary objects, net translations can also occur from cycle to cycle. These translations can be compensated for with methods for positioning, as in Section 3.

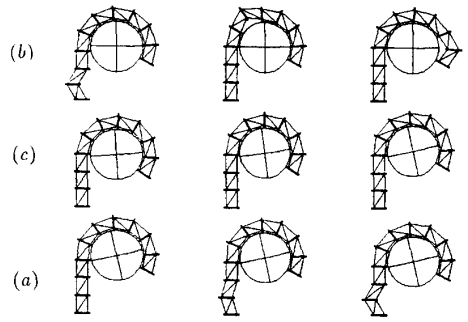

Figure 4: Hyper-Redundant Grasping of a Cylinder

This scheme is illustrated with a planar VGT manipulator that manipulates a disk. The backbone curve is nonextensible, and thus a pure bending grasping wave shape must be selected. For simplicity, a triangular grasping wave is used, employing the nondifferentiable three mode example in Section 3. This wave which will travel along the backbone curve and must track the boundary of the disk. The distance from the base of the manipulator to the rear of the wave as measured along the backbone curve is denoted by $s_{w_0}(t)$. The arclength of the traveling wave is $s_{w_1}(t) - s_{w_0}(t) = L$. The triangle wave has the curvature function:

$$\kappa_w(s,t) = a_1\delta(s - s_{w_0}) + a_2\delta\left(s - \frac{1}{2}(s_{w_0} + s_{w_1})\right) + a_3\delta(s - s_{w_1}) \quad (14)$$

where $\{a_1, a_2, a_3\}$ are chosen so that the grasping wave tracks the disk boundary according to the closed form inverse kinematics which result from this choice of curvature function, while $s_{w_0} = s_{w_0}(t)$ controls the grasping wave progression. In the

first grasping phase, a portion of the non-contact segment contracts to form a wave while tracking a line. During this phase, $a_1 = a_3 = \hat{\alpha}$, $a_2 = -2\hat{\alpha}$, where $\hat{\alpha} = \hat{\alpha}(t)$ specifies the slope of the wave. At the beginning of the wave formation, $\hat{\alpha}(t_0) = 0$, and at the end, $\hat{\alpha}(t_1) = \alpha$. For this wave shape, s_L is constant at $s_L = (1 - \cos\alpha)L$ throughout the second grasping phase.

Assume that the center of the disk is located at (r_c, h_c) measured in the x-y manipulator base frame. r_c is the disk radius and $h_c(t)$ is the y coordinate of the disk center, which varies in the first and third grasping phases. Let $s_c(t)$ denote the arclength along the backbone curve such that $y(s_c(t)) = h_c(t)$. If any part of the wave is in the interval $s \in [0, s_c]$, then $h_c < s_c$, otherwise $h_c = s_c$.

The disc boundary and the straight noncontact section form a *terrain curve* for the traveling wave to track [5]. The $\{a_i\}$ in (14) are determined so that the grasping wave locally tracks the 'terrain'. This is done by solving the inverse kinematics for the grasping wave segment as if it were a manipulator with curvature function (14) and effective end-effector location derived by taking the difference in position between the front of the wave ($s = s_{w_1}$) and the back of the wave ($s = s_{w_0}$). Similarly, the orientation is the angle between the tangents to the back and front of the wave. These are:

$$x_w(t) = T_1(s_{w_1}(t) - s_L, t) - T_1(s_{w_0}(t), t) \tag{15a}$$

$$y_w(t) = T_2(s_{w_1}(t) - s_L, t) - T_2(s_{w_0}(t), t) \tag{15b}$$

$$\theta_w(t) = \cos^{-1}\left[\frac{\partial \overline{T}}{\partial \nu}(s_{w_1}(t) - s_L, t) \cdot \frac{\partial \overline{T}}{\partial \nu}(s_{w_0}(t), t)\right] \tag{15c}$$

where $\overline{T}(\nu, t) = [T_1(\nu, t), T_2(\nu, t)]^T$ is the terrain curve, and ν is arclength measured along the terrain. For this example

$$\overline{T}(\nu, t) = W(\nu, 0, h_c(t))[0, \nu]^T + W(\nu, h_c(t), 1) \times \left[h_c(t) + \frac{1}{r_c}\sin\left(\frac{\nu - h_c(t)}{r_c}\right), h_c(t) + \frac{1}{r_c}\left(1 - \cos\left(\frac{\nu - h_c(t)}{r_c}\right)\right)\right]^T \tag{16}$$

which consists of a straight section for $0 \le \nu < h_c$, and a circular arc for $h_c \le \nu \le 1$. Again, s_L stands for the stride length (or slack) associated with the traveling wave segment as measured along the cylinder.

The first two phases are implemented by different behaviors of the functions s_{w_0} and a_i. In the third phase, the grasping wave has exited the end of the manipulator. The curvature function assumes the form:

$$\kappa_3(s, t) = W(s, s_c, 1)/r_c \tag{17}$$

where $s_c(t) = h_c(t)$ throughout this phase. This curvature function implements the 'unwrapping' of the object as $h_c(t)$ increases.

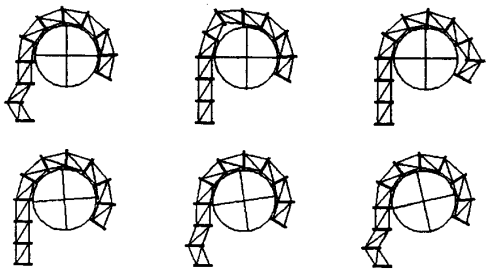

Figure 5: Turning an object fixed to an Axle

The above procedure can be altered so that no translation occurs. This is achieved by combining the first and third phases so that the translations of the object which occurred between phases cancels. This is shown in Figure 5. Symmetries of the grasping process can be exploited for dual arm hyper-redundant grasping and manipulation of objects as seen in Figure 6. See [2] for details.

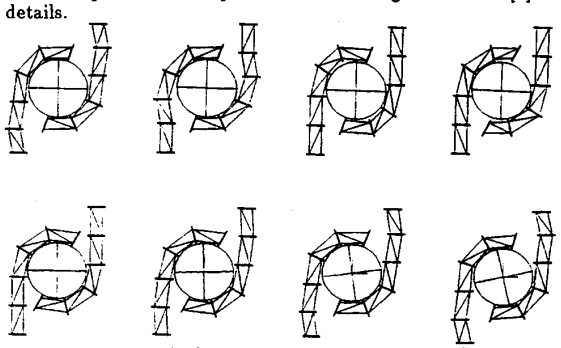

Figure 6: Dual Arm Grasping of a Cylinder

6. Design and Implementation of a Hyper-Redundant Robot

To demonstrate and validate the kinematic modeling methods, obstacle avoidance, locomotion, and grasping schemes developed in [2-7], we have developed a 30 degree-of-freedom planar hyper-redundant robot. The robot has a VGT morphology, consisting of 10 identical three degree-of-freedom truss modules. Each truss module (Figure 7) has 3 prismatic joints, which are actually implemented with D.C. servo motors and ball screw drives. The actuators can vary in length from a minimum of 12 inches to a maximum of 18 inches, and can generate a force of 75 points during motion, and 225 pounds staticly.

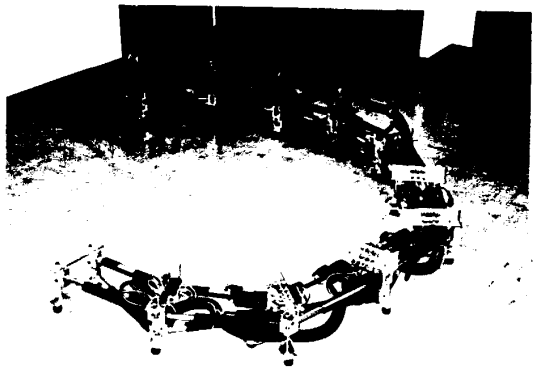

Figure 7: 30 Degree-of-Freedom Hyper-redundant Robot

The robot is designed to demonstrate both fixed base manipulation and obstacle avoidance as well as traveling and standing wave locomotion [5]. For manipulation and grasping experiments, castors are attached to the fixed elements of the modules in order to allow low friction motion over the laboratory floor, which is the plane of manipulation. For locomotion experiments, the manipulator is detached from its base, and turned on its side. Each module is equipped with rubber faced ratchet wheels which permit rolling in only one direction, which is required for

standing wave locomotion [5]. For traveling wave locomotion, the wheels are locked in place, and the wheels' rubber surfaces are use as passive feet. These rubber surfaces are also used as the contact points for planar grasping operations. Further, because of its modular shape, the robot can be easily separated into two 15 degree-of-freedom manipulators for dual arm manipulation experiments, such as in Figure 6.

Let the length of a module (from fixed element to fixed element) divided by the width of the module (distance between actuator pivot points) be called the *module aspect ratio*. The fixed elements of the truss modules are designed so that each module's aspect ratio can be varied for different applications. A very high module aspect ratio leads to greater dexterity, but has reduced mechanical advantage. High aspect ratios are advantageous for obstacle avoidance in tightly constrained spaces. Conversely, low aspect ratios provide greater mechanical strength at the expense of dexterity.

The robot system is controlled by an a multiprocessing computer, which consists of Heurikon 68030 processing boards in a VME bus, which is in turn connected to a Sun Microsystems 4/260 computer. The WindRiver Systems VxWorks software system provides operating system and system management functions. The 68030 processors compute the kinematic, path planning, sensor processing, and task level coordination functions. The Sun computer is used primarily for user interface and data storage.

To date, hyper-redundant manipulators have remained largely a laboratory curiosity. There are a number of reasons for this: (1) previous kinematic modeling techniques have not been particularly efficient or well suited to the needs of hyper-redundant robot task modeling; (2) the mechanical design and implementation of hyper-redundant robots has been perceived as unnecessarily complex; and (3) hyper-redundant robots are not anthropomorphic, and pose interesting programming problems. The authors have been working to overcome (1), and in some cases hyper-redundant mechanism design is not unreasonably complex. (3) remains a challenge.

As demonstrated by our design, hyper-redundant robots do have some inherent design and operating advantages. First, hyper-redundant robot design is often amenable to simple modular implementation. Second, in most operational scenarios, only a small fraction of the robot actuators are actually in use, minimizing their duty cycle. Further, such systems are highly fault tolerant, capable of sustaining numerous actuator failures. Such systems can also be weight efficient. The 30 degree-of-freedom robot can contract to a minimum length of 12 feet, and extend to a maximum of 18 feet, but weighs less than 55 kg. Of course, hyper-redundant robots have obvious disadvantages, and are best suited to niche applications. Small errors in each actuator position can accumulate into large base-to-tip position errors. Thus, hyper-redundant manipulators are at a disadvantage for precise positioning tasks, unless they are instrumented with numerous sensors and equipped with sensor-based control strategies which adjust for local errors. This is a subject for future research.

Conclusions

In the past, little work has been done in applying hyper-redundant robots to practical situations because of the complex kinematics of even the most idealized models. Using recently developed methods, the kinematic complexity has been overcome and practical application of hyper-redundant robots can be considered.

Aknowledgements

This work was sponsored in part by a NASA Graduate Student Researchers Program fellowship (for the first author), and the Caltech President's Fund, grant #PF-331.

References

[1] Anderson, V.V., Horn, R.C., "Tensor-arm Manipulator design," ASME Trans., Vol. 67-DE-57, pp. 1-12, 1967.

[2] Chirikjian, G.S., "Theory and Applications of Hyper-Redundant Robotic Mechanisms," Ph.D Dissertation, Department of Engineering and Applied Science, California Institute of Technology, (to appear spring 1992).

[3] Chirikjian, G.S., Burdick, J.W., "An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators," *1990 IEEE International Conference on Robotics and Automation*, Cincinnati, OH,May 13-18, 1990.

[4] Chirikjian, G.S., Burdick, J.W., "The Kinematics of Hyper-Redundant Manipulators," *Proceedings of the 21st Biennial ASME Mechanisms Conference*, Chicago, Illinois, September, 1990.

[5] Chirikjian, G.S., Burdick, J.W., "Kinematics of Hyper-Redundant Locomotion with Applications to Grasping," *1991 IEEE Conference on Robotics and Automation*, Sacramento, CA, April, 1991.

[6] Chirikjian, G.S., Burdick, J.W., "Parallel Formulation of the Inverse Kinematics of Modular Hyper-Redundant Manipulators," *1991 IEEE Conference on Robotics and Automation*, Sacramento, CA, April, 1991.

[7] Chirikjian, G.S. and Burdick, J.W., "On the Determination of Kinematically Optimal Hyper-Redundant Manipulator Configurations," *Robotics and Mechanical Systems Report* no. RMS-91-02, September, 1991.

[8] Hayashi, A., Park, J., Kuipers, B.J., "Toward Planning and Control of Highly Redundant Manipulators," *Fifth IEEE International Symposium on Intelligent Control, 1990.*

[9] Hirose, S., Umetani, Y., "Kinematic Control of Active Cord Mechanism With Tactile Sensors," *Proceedings of Second International CISM-IFT Symposium on Theory and Practice of Robots and Manipulators*, pp. 241-252, 1976.

[10] Hirose, S., Ma, S., "Coupled Tendon-driven Multijoint Manipulator," *1991 IEEE Conference on Robotics and Automation*, Sacramento, CA, April, pp. 1268-1275, 1991.

[11] Jain, S., Kramer, S.N., "Forward and Inverse Kinematic Solution of the Variable Geometry Truss Robot Based on an N-Celled ' Tetrahedron-Tetrahedron Truss," *Journal of Mechanical Design*, Vol. 112, pp. 16 - 22, March, 1990.

[12] Klein, C.A., Huang, C.H., "Review of the Pseudoinverse for Control of Kinematically Redundant Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, March 1983.

[13] Naccarato, F., Hughes, P.C., "An inverse Kinematics Algorithm for A Highly Redundant Variable-Geometry-Truss Manipulator," *Proceedings of the 3rd Annual Conference on Aerospace Computational Control*, Oxnard, CA, Dec. 15, 1989.

[14] Pettinato, J.S., Stephanou, H.E., "Manipulability and Stability of a Tentacle Based Robot Manipulator," *IEEE Conference on Robotics and Automation*, May 15-19, 1989, Scottsdale, AZ, pp. 458-463.

[15] Salerno, R.J., Reinholtz, C.F., Robertshaw, H.H., "Shape Control of High Degree-of-Freedom Variable Geometry Trusses," *Workshop on Computational Aspects in the Control of Flexible Systems, part 2*, Williamsburg, VA, July 12-14, 1988.

[16] Suzumori, K., Iikura, S., Tanaka, H., "Development of Flexible Microactuator and It's Applications to Robotic Mechanisms," *1991 IEEE Conference on Robotics and Automation*, Sacramento, CA, April, 1991.

[17] Wilson, J.F., Mahajan, U., "The Mechanics and Positioning of Highly Flexible Manipulator Limbs," *Journal of Mechanisms, Transmissions, and Automation in Design*. Vol, 111, June 1989.